

## Øving III

### Oppgave 1

- a) Hva er forskjellen på et høynivå- og lavnivå-programmeringsspråk?

Jo nærmere man kommer maskinkode, jo lavere regnes nivået på programmeringsspråket å være. Assembly er et eksempel på et veldig lavnivå språk. Koden som skrives her er veldig nærme maskinkoden som CPUen leser for å utføre programmet. Å skrive lavnivå-programmeringsspråk krever at man vet hvordan CPUen fungerer.

Mange nye språk er høynivå-programmeringsspråk og gjør det mulig å gjøre avanserte og komplekse oppgaver uten å skrive for mange kommandoer. Funksjonene her er også vanligvis mye nærmere det normale skriftspråket (engelsk) enn man ser i lavnivå-programmeringsspråk.

Når man skal kjøre høynivå-programmeringsspråk krever det mer for å gjøre det om til maskinkode (kompilere), samtidig er de vanligvis også tregere enn lavnivå-programmeringsspråk.

- b) Hva gjør programtelleren (program counter)?

Programtelleren eller instruksjonspekeren er et spesielt prosessorregister som brukes for å angi hvor langt prosessoren har kommet i utføringen av en instruksjonssekvens. Den angir enten adressen til instruksjonen som utføres eller adressen til neste instruksjon som skal utføres.

For hver instruksjon som utføres, så avanserer vanligvis programtelleren én instruksjon fremover i minnet. Unntak fra denne regelen er ved prosedyrekall, hopp og forgrening av instruksjonsflyt. I disse tilfellene blir en ny minneadresse lastet inn i programtelleren, noe som får prosessoren til å begynne å hente instruksjoner fra en annen del av minnet.

- c) Hva gjør en kompilator?

En kompilator gjør om et program fra et høynivå-programmeringsspråk til maskinkode. Til forskjell fra interpreter kjøres ikke programmet med en gang. Når et program er kompilert vil man ha programmet som en egen fil tilgjengelig. C, C++ og C# er eksempler på språk hvor dette skjer. Etter at programmet er kompilert må man starte programmet ved å kjøre .exe-fila. Dette krever ikke å måtte kompilere programmet på nytt.

### Oppgave 3

- c) Argument er det funksjonen sender eller mottar. F.eks x, number eller value. Parameteret er den verdien argumentet inneholder. Boken definerer det som "parameter er en spesiell variabel som er gitt verdien til et argument når funksjonen kalles".
- d)

e) Variabelen a settes i det globale scopet når det er definert.

Funksjonen foo kalles med 2 som parameter for argumentet a. I funksjonen foo blir a opphøyd i andre og printet. Det printes da **4** til konsollen. Fordi dette er inne i funksjonen er dette lokale variabler som ikke påvirker variabelen satt i det globale scopet.

a printes i det globale scopet, denne har ikke blitt endret siden linje 1 og det skrives derfor ut 1.